CLAIMS

What is claimed is:

1. A method for transparency rendering in a graphics pipeline, comprising:

(a) collecting colored-transparency information from a plurality of depth layers in a scene to be rendered;

(b) storing the collected colored-transparency information in memory; and

(c) blending the colored-transparency information from the depth layers in a predetermined order.

2. The method as recited in claim 1, wherein the colored-transparency information is collected from at least two depth layers.

3. The method as recited in claim 1, wherein the colored-transparency information is stored in a plurality of texture maps.

4. The method as recited in claim 3, wherein each of the texture maps corresponds with one of the depth layers.

5. The method as recited in claim 4, wherein the texture maps are stored in memory.

6. The method as recited in claim 1, and further comprising rendering opaque objects in the scene.

7. The method as recited in claim 6, the opaque objects in the scene are rendered prior to blending the colored-transparency information therewith.

8. The method as recited in claim 1, wherein the memory includes a frame buffer.

9. The method as recited in claim 1, wherein the blending includes linear blending.

10. The method as recited in claim 1, wherein the colored-transparency information is collected utilizing depth peeling.

11. The method as recited in claim 10, wherein the depth peeling includes executing a first rendering pass for collecting colored-transparency information relating to a first depth layer, and executing additional rendering passes for collecting additional colored-transparency information relating to additional depth layers.

12. The method as recited in claim 11, wherein the first rendering pass produces a shadow map relating to the first depth layer.

13. The method as recited in claim 11, wherein a shadow-mapping feature is enabled during the additional rendering passes for defining a previous depth layer.

14. The method as recited in claim 11, wherein the additional rendering passes are taken from the same eye position from which the first rendering pass is taken.

15. The method as recited in claim 1, wherein the colored-transparency information is collected utilizing depth peeling including executing a first rendering pass for generating a shadow map from which first colored-transparency information relating to a first depth layer is collected, and executing additional rendering passes with a shadow-mapping feature enabled and from the same eye position from which the first rendering pass is

taken for collecting additional colored-transparency information relating to additional depth layers.

16. The method as recited in claim 15, wherein the additional colored-transparency information relating to the additional depth layers is collected by removing a portion of the scene associated with a previous depth layer.

17. The method as recited in claim 16, wherein the additional colored-transparency information relating to the additional depth layers is collected by performing a test to determine which portion of the scene to remove.

18. The method as recited in claim 17, wherein the test determines whether the portion of the scene is behind the previous depth layer.

19. The method as recited in claim 18, wherein the portion of the scene is removed upon the test determining that the portion of the scene is behind the previous depth layer.

20. The method as recited in claim 19, wherein the test calculates a difference between a previous z-value relating to the previous depth layer and a present z-value relating to one of the additional depth layers.

21. The method as recited in claim 20, wherein the portion of the scene is removed upon no difference being calculated between the previous z-value relating to the previous depth layer and the present z-value relating to one of the additional depth layers.

22. The method as recited in claim 21, wherein the z-values relating to all depth layers are produced with the same interpolation-related method for improving an accuracy of the test.

23. A computer program product for transparency rendering in a graphics pipeline, comprising:

(a) computer code for collecting colored-transparency information from a plurality of depth layers in a scene to be rendered;

(b) computer code for storing the collected colored-transparency information in memory; and

(c) computer code for blending the colored-transparency information from the depth layers in a predetermined order.

24. A system for transparency rendering in a graphics pipeline, comprising:

(a) logic for collecting colored-transparency information from a plurality of depth layers in a scene to be rendered;

(b) memory for storing the collected colored-transparency information; and

(c) a renderer coupled to the memory for blending the colored-transparency information from the depth layers in a predetermined order.

25. A system for transparency rendering in a graphics pipeline, comprising:

(a) logic for collecting colored-transparency information from a plurality of depth layers in a scene to be rendered;

(b) memory for storing the collected colored-transparency information; and

(c) register combiners coupled to the memory for blending the colored-transparency information from the depth layers in a predetermined order.

26. A method for transparency rendering in a graphics pipeline, comprising:

(a) collecting colored-transparency information from at least two depth layers in a scene;

(b) storing the collected colored-transparency information in the form of a plurality of texture maps;

(c) rendering the opaque objects in the scene;

(d) storing the rendering of the opaque objects in memory;

(e) identifying one of the depth layers to be blended;

(f) blending the colored-transparency information from the identified depth layer with contents of the memory utilizing a corresponding one of the texture maps;

(g) storing results of (f) in the memory; and

(h) repeating acts (e)-(g).

27. A computer program product for transparency rendering in a graphics pipeline, comprising:

(a) computer code for collecting colored-transparency information from at least two depth layers in a scene;

(b) computer code for storing the collected colored-transparency information in the form of a plurality of texture maps;

(c) computer code for rendering opaque objects in the scene;

(d) computer code for storing the opaque object in memory;

(e) computer code for identifying one of the depth layers to be blended;

(f) computer code for blending the colored-transparency information from the identified depth layer with contents of the memory utilizing a corresponding one of the texture maps;

(g) computer code for storing results of (f) in the memory; and

(h) computer code for repeating acts (e)-(g).

28. A method for transparency rendering in a graphics pipeline, comprising:

(a) collecting colored-transparency information from a plurality of depth layers in a scene to be rendered by:

(i) executing a first rendering pass for generating a shadow map and for collecting first colored-transparency information relating to a first depth layer, and

(ii) executing additional rendering passes with a shadow-mapping feature enabled and from the same eye position from which the first rendering pass is taken for generating additional shadow maps and for collecting additional colored-transparency information relating to additional depth layers;

(b) storing the collected colored-transparency information in memory; and

(c) blending the colored-transparency information from the depth layers.

29. A computer program product for transparency rendering in a graphics pipeline, comprising:

(a) computer code for collecting colored-transparency information from a plurality of depth layers in a scene to be rendered by:

(i) executing a first rendering pass for generating a shadow map and for collecting first colored-transparency information relating to a first depth layer, and

(ii) executing additional rendering passes with a shadow mapping feature enabled and from the same eye position from which the first rendering pass is taken for generating additional shadow maps and for collecting additional colored-transparency information relating to additional depth layers;

(b) computer code for storing the collected colored-transparency information in memory; and

(c) computer code for blending the colored-transparency information from the depth layers.